1.Git config –global user.name “github username”

2. Git config –global user.email “github email”

1 and 2 commands are to configure git hub account with git

3. git config --global -e = to see or edit the information available in global

4. git config – local -e == to see or edit local information

5. git config –system -e = to see or edit system information

6. cd =to change directory

7. mkdir = to create new directory using git bash

8. cd .. == to come backward from one directory to another

9.:q = to come out of vi editor

10. alias npp=”notepad+.exe -multiInst -nosession to create short cut for notepad++ as npp

11. git config –global alias.shortcutname “git command”= to create new shortcut name

12. clear= to clear all above

13. ls= to see list of files in that directory

14. ls -al = to see hidden files like .git

15.git config –global core.editor “notepad++.exe -multiInst -nosession” = to make notepad++ as editor

16. git add file name = to sent given filename to staging area

17. git add . == to sent all the files in directory to staging area

18.git init= to initiate git into local directory

19. git clone repository url = to clone from given url with that repository name

20. git clone repository url filename = to clone from url with the given filename

21. git push origin master = to push to github repository { when we clone from github}

22. git remote add origin url = to add given url as origin { when we sent files from local system to github}

23. git push -u origin master= to push to above url same as above

24.ntepad++ filename = to create new file in that directory with filename

25. git status= to check the status of files

26.notepad++ .gitignore = to create folder , if we enter filename inthat git cannot access to that it ignores that file.

27. \*.txt = if we use this in .gitignore every file with .txt files gets ignored

28. git diff == to see difference in file in modified and working directory file

29.git diff –staged = to see what files are in staged position

30. git commit = to commit it will opens notepad so we need to write description about that commit

31. git commit -m “ msg “ == inline commit

32. git commit -v == we can see what commit is going on

33.git log= to see history of that file

34. git log -p = to see the difference between each commits

35. git log -4 == to see only difference between commits

36. git log –pretty=oneline = to see all commits in oneline

37. git branch branchname = to create branch with given branch name

38. git checkout branch name= to go to that newly created branch

39. git checkout -b branch name= it creates new branch and goes to that new branch

40 git branch -a ==to see what branches are available

41. git merge branchname = to merge one branch into another

42. cat filename = to see total file

43. git branch -d branchname = to delete branch

44.git branch –merge == to see what branches are merged

45.git fetch origin master= to pull changes from github

46.git merge origin master= to merge the changes from fetch

47. git pull origin master=to fetch and merge the changes

48. git difftool = to see differences visually

49. git mergetool = to merge the conflicts visually

50. git rebase branchname = to merge the branch into another same as git merge

51.git log –oneline –graph –all = to see all the commits with graph in single line

52. git stash = to delete the modified file

53. git stash list = to see what files are in stash

54. git stash apply = to apply latest stash file to our working directory

55. git stash drop= to drop stash file

56. git stash pop= it apply and drop that file

57. git stash apply stash @{1} = it applies stash 1

58.git clean -f = to remove untracked files

59. git clean -f-d= to remove folders untracked

60. git clean -f -n = to dryrun the files

61. git tag tagname= to make tag to latest commit

62.git tag –list == to see list of tags

63. git tag -a tagname = to write description to that tag

64.git tag -a tagname -m “ msg” = inline tag msg

65.git diff tagname 1 tagname 2 == to see difference between the two tag levels

66. git tag -d tagname = to delete tag

67. git tag -a tagname commitid = to give tag to given commit id

68. git tag tagname 1-f commitid = to remove tag name for tag name 1 and give that tag to given commit id

69.git push origin tag name = to push that given tag name to origin

70. git push origin master –tags = to push all tags to origin master

71. git commit -am “msg” = it will directly adds and commits the file